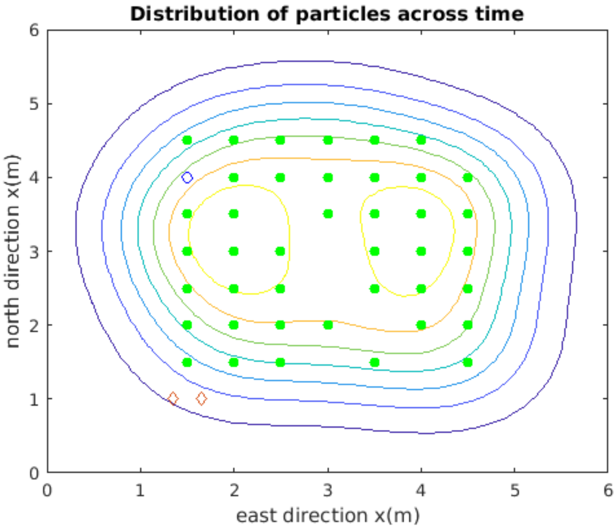
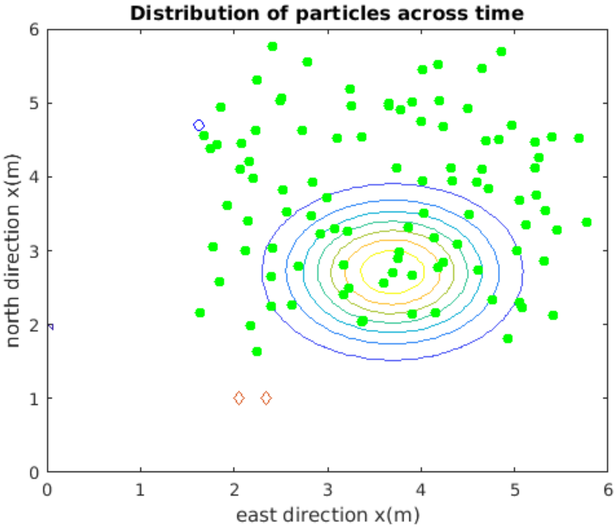
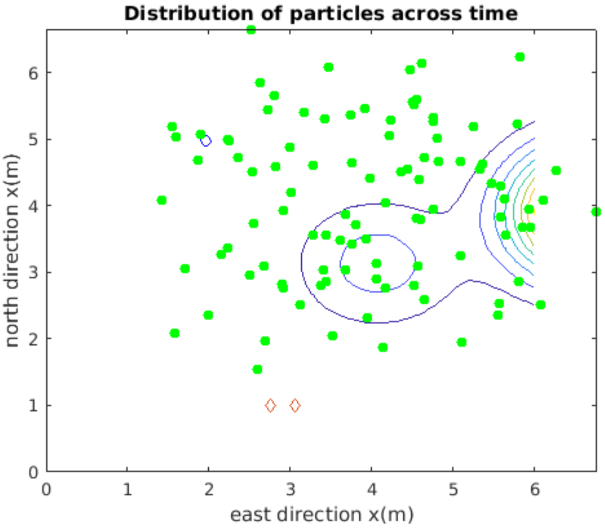
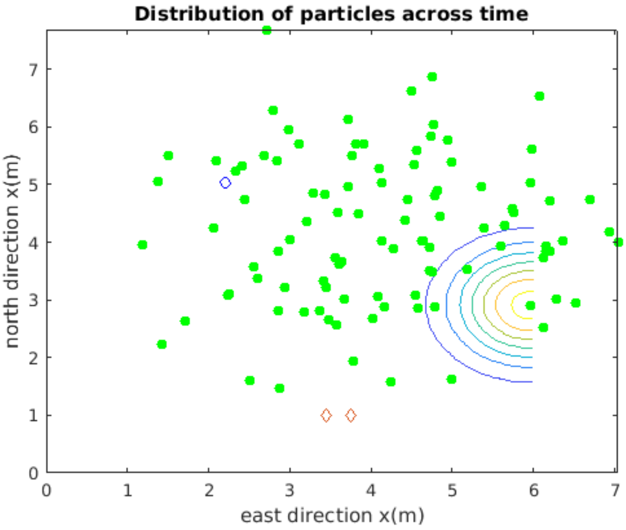
```
clear;clc;
%Source Motion Model Parameters
delta_t = 0.375;
N_particles = 100;
N_steps = int64(2/delta_t);
beta = 2;
v_bar = 1;
frame_length = 0.05;
a = exp(-beta*delta_t);
as = exp(-beta*frame_length);
b = v_bar*sqrt(1-a^2);
bs = v_bar*sqrt(1-as^2);
F = [eye(2), a*delta_t*eye(2);zeros(2,2),a*eye(2)];
Fs = [eye(2), as*frame_length*eye(2);zeros(2,2),as*eye(2)];
Q = [b^2*delta_t^2*eye(2),zeros(2,2);zeros(2,2),b^2*eye(2)];
Qs = [bs^2*frame_length^2*eye(2),zeros(2,2);zeros(2,2),bs^2*eye(2)];


%True source and microphone motion models
source = zeros(N_steps+1,4);
source(1,:) = 0.5*randi([3 9],1,4) ;
mic = zeros(N_steps+1,2,3);
mic(1,1,:) = [1.35,1,1.5];
mic(1,2,:) = [1.65,1,1.5];
mic(end,1,:)  = [4.85,1,1.5];
mic(end,2,:) = [5.15,1,1.5];

for i = 1:3
        mic(:,1,i) = linspace(mic(1,1,i),mic(end,1,i),N_steps+1);
        mic(:,2,i) = linspace(mic(1,2,i),mic(end,2,i),N_steps+1);
end

%rir parameters and model
room_dim = [6 6 2.5];
fs = 8000;
c = 340;
rt60  = 0.5;
%rir(1,:,:) = rir_generator(c,fs,mic(1,:,:),[source(1,1:2) 0],room_dim,rt60,rir_samples);

%Sampled Source Positions
source_samp = zeros(N_steps+1,N_particles,4);
source_samp(1,:,:) = 0.5*randi([3 9],N_particles,4);


%Grid points
[X,Y] = meshgrid(0:0.1:6,0:0.1:6);
grid_pts = [X(:),Y(:)];
kdeprob = zeros(N_steps,size(grid_pts,1));

%Signal Array : sig
signal_raw = load('timit_audio.mat');
signal_raw = signal_raw.audio_samps;
signal_raw = signal_raw(1:16000);
%signal = reshape(signal_raw(1:(N_steps-1)*fs*delta_t), [N_steps-1 fs*delta_t]);
%signal(N_steps,:) = reshape(signal_raw((N_steps-1)*fs*delta_t +1 :end),[1 (fs*10 -(N_steps-1)*fs*delta_t)]);

%STFT params
window_size = 400;%Frame length
window = rectwin(window_size);
n_bins = 2^nextpow2(window_size);
outside_source  = 0;
outside_samples = 0;



%weights
w = zeros(N_steps+1,N_particles);
w(1,:) = rand(1,N_particles)';
for t = 1:N_steps
        temp_source = mvnrnd(F*reshape(source(t,:),[4 1]),Q,1);
    a = temp_source(1:2) <0 ;
    b = temp_source(1:2) > 6;
    outside_source = outside_source + max(sum(a(:)),sum(b(:)));
        %while ((sum(a(:)) > 0) || (sum(b(:)) > 0))
        %       temp_source = mvnrnd(F*reshape(source(t,:),[4 1])   ,Q,1);
    %end

        source(t+1,:) = temp_source;
    if t == N_steps
        Y = stft(signal_raw(fs*(N_steps-1)*delta_t+1:end),window,window_size,n_bins,fs);
        rir = rir_generator(c,fs,reshape(mic(t,:,:),[2 3]),[source(t,1:2) 0],room_dim,rt60,size(Y,1));
    else
        Y = stft(signal_raw((t-1)*fs*delta_t+1: t*fs*delta_t ),window,window_size,n_bins,fs);
        rir = rir_generator(c,fs,reshape(mic(t,:,:),[2 3]),[source(t,1:2) 0],room_dim,rt60,size(Y,1));
    end
    Y = Y';
    w(t+1,:) = w(t,:);
    source_samp(t+1,:,:)=source_samp(t,:,:);
    for j=1:size(Y,1)
        for j = 1:N_particles
            temp_samp = mvnrnd(Fs*reshape(source_samp(t+1,j,:),[4,1]),Qs,1);
            a = temp_samp(:,1:2) < 0;
            b =  temp_samp(:,1:2) > 6;
            outside_samples = outside_samples + max(sum(a(:)),sum(b(:)));
            %while ( (sum(a(:)) > 0) || (sum(b(:)) > 0))
            %   temp_samp = mvnrnd(F*reshape(source_samp(t,j,:),[4,1]),Q,1);
            %end
            source_samp(t+1,j,:) = temp_samp;
        end
        S1 = Y(t,:).*rir(1,:);
        S2 = Y(t,:).*rir(2,:);
        S1_ind = double(S1~=0);
        S2_ind = double(S2~=0);
        tot_ind = S1_ind + S2_ind;
        S1 = S1(tot_ind==2);
        S2 = S2(tot_ind==2);

        prob_new = SSP_EM(reshape(mic(t,:,:),[2 3]),reshape(source_samp(t+1,:,:),[N_particles 4])',[S1;S2],10e-1);
        w(t+1,:) = w(t+1,:).*prob_new;
        w(t+1,:) = w(t+1,:)./sum(w(t+1,:));
        %Update equations for weight
    end
    figure;
    ksdensity(reshape(source_samp(t,:,1:2),[N_particles,2]),grid_pts,'Weights',reshape(w(t,:,:),[N_particles,1]),'PlotFcn','contour');
    hold on;
    scatter(source(t,1),source(t,2),'bo','DisplayName','Source Position');
    hold on;
    scatter(mic(t,:,1),mic(t,:,2),'d','DisplayName','Mic positions');
    hold on;
    scatter(source_samp(t,:,1),source_samp(t,:,2),'go','filled','DisplayName','Sampled Source Positions');
    xlabel('east direction x(m)');
    ylabel('north direction x(m)');
    title('Distribution of particles across time');
    legend;
end
disp("Number of source points outside room");
disp(outside_source);
disp("Number of sampled source points outside the room");
disp(outside_samples);
```

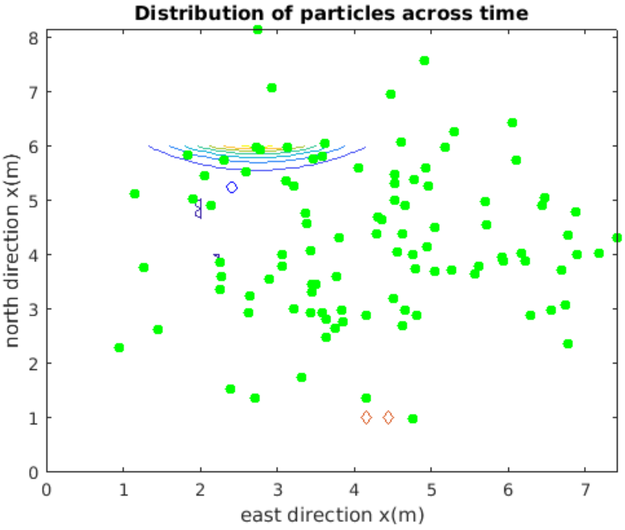
```
Room Impulse Response Generator (Version 2.1.20141124) by Emanuel Habets
Copyright (C) 2003-2014 E.A.P. Habets, The Netherlands.
Room Impulse Response Generator (Version 2.1.20141124) by Emanuel Habets
Copyright (C) 2003-2014 E.A.P. Habets, The Netherlands.
Room Impulse Response Generator (Version 2.1.20141124) by Emanuel Habets
Copyright (C) 2003-2014 E.A.P. Habets, The Netherlands.
Room Impulse Response Generator (Version 2.1.20141124) by Emanuel Habets
Copyright (C) 2003-2014 E.A.P. Habets, The Netherlands.
Room Impulse Response Generator (Version 2.1.20141124) by Emanuel Habets
Copyright (C) 2003-2014 E.A.P. Habets, The Netherlands.
Number of source points outside room
     0

Number of sampled source points outside the room
   575
```


Distribution of particles across time
north direction x(m)
east direction x(m)
0
1
2
3
4
5
6


Distribution of particles across time
north direction x(m)
east direction x(m)
0
1
2
3
4
5
6


Distribution of particles across time
north direction x(m)
east direction x(m)
0
1
2
3
4
5
6





Published with MATLAB® R2017a